

Service Casting: A Proposal for Advertising Web Services

<http://sciflo.jpl.nasa.gov/scast>



Brian Wilson
Jet Propulsion Laboratory
Brian.Wilson@jpl.nasa.gov

**Advertise Web Services via Atom
feeds, enabling simple aggregation,
discovery, & machine invocation.**





Motivation & Goals

- Reduce advertising services to publishing an “scast” (Atom) feed
 - Feed readers already aggregate & filter feeds
 - Search also available
- Decentralized repository with auto-aggregation
 - Service providers control publishing of advertisements
 - Scast discoverable in the cloud
 - Publish scast to Federation list by creating wiki link
 - Script auto-aggregates to list of scasts (OPML file)
- Scast contains machine-readable metadata to categorize services and enable auto-invocation
 - Links to interface, service endpoint, human documentation
- So simple it will be adopted (unlike previous attempts).
 - Make free tools available
 - Evangelize





Service Casting Quick Summary

- **Advertise bundle of Web Services via Atom Feed**
 - Service Casts (scast) from many providers
 - With links pointing to callable interface (WSDL) & docs.
 - Service provider pushes new ads when services change, or periodically
- **Simple Aggregation & Display**
 - All feed readers aggregate multiple feeds & import OPML
 - Publish bundled ads on Federation web site as OPML file
 - Display feeds in HTML or custom feed reader
- **Discovery**
 - Can search for feeds in Google FeedReader
 - Provide 'opensearch' interface that returns scasts
- **Machine Readable => Auto Service Invocation**
 - SOAP/REST service interfaces described by WSDL
 - Or REST interfaces described by WADL
 - Or known interfaces for OGC WMS/WCS/WFS



Browsing Service Casts

In Specialized Feed Reader / Aggregator

Signed in as bdwilson | [logout](#)

Feeds

Add Feed Remove

My Feeds

SciFlo Data Query/Access Services (df3)

- RGIS Base Imagery Service
- ADaM Services: Data mining and image process
- QuikSCAT Hurricane Ultra High-Res Winds
- GHRSSST AMSR-E SST
- ADaM Mining Web Services: Optimization Algori
- ADaM Mining Web Services: Image Processing
- ADaM Services: Data mining and image process
- HD - NASA's Jet Propulsion Laboratory
- NSIDC Atlas of the Cryosphere
- RGIS Base Imagery Service
- NSIDC Snow and Ice Cover
- ADaM Mining Web Services: Utility Services
- ADaM Mining Web Services: Texture Algorithms
- ADaM Mining Web Services: Pattern Recognition
- ADaM Mining Web Services: Domain Algorithms
- OOSTethys/MMI (Marine Metadata Interoperabilit
- Freeze/Thaw Earth System Data Record Data Vi

SciFlo Data Query/Access

Open All

Reading Pane

Summary

Title

Date ▾

GeoRegionQuery

Brian Wilson

3/13 1:32 am

Space/time query and granule URL lookup services for multiple EOS satellite datasets at Levels L2/L3: AIRS, MODIS, MISR, GPS, and AERONET (ground network).

FindDataById

Brian Wilson

3/13 1:32 am

Translate a list of SciFlo data granule IDs to a list of on-line URLs.

View in New Tab

Go to Post

GeoRegionQuery

by Brian Wilson

Mar 13, 2008, 1:32 am

Example Call

```
GeoRegionQuery(dataSetId='AIRS', level='L2', version=None, startTime='2006/01/01T00:00:00', endTime='2006/02/01T00:00:00', latMin=-90., latMax=90., lonMin=-180., lonMax=180., responseGroups='Large' )
```

[Service interface description](#) [Server endpoint](#) [Service documentation](#) [SCast Source](#)





Why Atom feed instead of RSS?

- Multiple Typed Links possible in each Entry
 - `<link rel="<purpose>" type="application/wsdl+xml" href="" />`
 - Links to interface, service endpoint, & human documentation
- Can embed tags from arbitrary namespaces
 - Define scast namespace, and register it
 - `<scast:serviceType>` = SOAP, REST, OGC.WMS, HUMAN
 - Specific services can embed more XML metadata (sciflo)
- Embed arbitrary content in 'typed' `<content>` tag
 - `<content type="xhtml" or type="xml">`
 - Use for more documentation, or more XML metadata
- Use `<category>` tag to characterize services
 - Use taxonomy of services, or just text keywords
- Opensearch and GoogleData standards extend Atom
- Atom got timestamps right (unlike RSS)
 - ISO YYYY-MM-DDTHH:MM:SS



Categorizing Service Semantics

- **Service Semantics**
 - OGC.WXS: WMS, WCS, WFS, etc.
 - OpenSearch, OpenDAP
 - AdHoc, Simple (for SOAP or REST services)
 - Human (e.g. Web apps & AJAX GUI's for humans)
 - [Your service type here]
- **Service Protocols (syntax)**
 - Tells machine how to interpret links
 - SOAP / WSDL - parse WSDL interface and call service
 - REST / WADL
 - For OGC, do GetCapabilities call
 - HTTP, HTTPPOST
 - AJAX, JavaWebStart – for Human GUIs
- **How do we further type SOAP/REST semantics?**
 - Use <category> to further specify semantics of call
 - What services are well known besides OGC.WXS?





Typed Links

- Service Cast Namespace
 - scast = <http://sciflo.jpl.nasa.gov/ServiceCast/2008v1>
 - Links have typed purpose (rel attribute)
- Rel = “scast:interfaceDescription”
 - URL pointing to WSDL, WADL, or WXS GetCapabilities call
 - Also the “default” link for each feed entry
 - For human-oriented GUI’s, this link probably the same as serviceEndpoint.
- Rel = “scast:serviceEndpoint”
 - URL pointing to where services can be called
 - For SOAP, also contained in WSDL file
- Rel = “scast:serviceDocumentation”
 - URL pointing to human-readable documentation
 - Also the “alternate” link for each feed entry
- Rel = what else?



Scast feed format

Just an Atom feed, with one entry for each advertised service

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xml:lang="en"
      xmlns:scast="http://sciflo.jpl.nasa.gov/serviceCasting/2009v1"
      xmlns:sf="http://sciflo.jpl.nasa.gov/2006v1/sf">
  <title>SciFlo Data Query/Access Services</title>
  <subtitle>On SciFlo node sciflo.jpl.nasa.gov</subtitle>
  <link rel="self" href="http://sciflo.jpl.nasa.gov/sciflo/web/feed.atom.xml"/>
  <link href="http://sciflo.jpl.nasa.gov/" />
  <updated>2008-03-13T01:32:02</updated>
  <author>
    <name>Brian Wilson</name>
    <email>sciflo@sciflo.jpl.nasa.gov</email>
  </author>
  <id>uri:http://scifo.jpl.nasa.gov/sciflo/v1/services/</id>

  <entry>

  <entry>
</feed>
```





Scast feed entry

```
<entry>
  <title>GeoRegionQuery</title>
  <id>uri:http://scifo.jpl.nasa.gov/sciflo/v1/services/GeoRegionQuery</id>
  <updated>2008-03-13T01:32:02</updated>
  <summary>Space-time query and granule URL lookup services for multiple EOS satellite
datasets at Level L2:  AIRS, MODIS, MISR, GPS, and AERONET (ground network).
</summary>
  <scast:serviceSemantics>Simple</scast:serviceSemantics>
  <scast:serviceProtocol>SOAP</scast:serviceProtocol>
  <category schema="scast" term="data query space time" />
  <link type="text/html" xml:lang="en-us"
    title="Service documentation"
    href="https://sciflo.jpl.nasa.gov/SciFloWiki/SpaceTimeQuery" />
  <link rel="scast:interfaceDescription" type="application/wsd+xml"
    title="Service interface description"
    href="http://sciflo.jpl.nasa.gov/sciflo/services/wsd/2006v1/EOSServices" />
  <link rel="scast:serviceEndpoint" type="application/soap+xml"
    title="Server endpoint"
    href="http://sciflo.jpl.nasa.gov/sciflo/services/soap" />
  <link rel="scast:serviceDocumentation" type="text/html" xml:lang="en-us"
    title="Service documentation"
    href="https://sciflo.jpl.nasa.gov/SciFloWiki/SpaceTimeQuery" />
  <link rel="alternate" type="application/wsd+xml"
    title="Service interface description"
    href=
"http://sciflo.jpl.nasa.gov/sciflo/services/soap?wsdl=http://-sciflo.jpl.nasa.gov/2006v1/sf/E
OSServices" />
```





Scast feed entry (2)

(rest of entry)

```
<content type="xhtml">
  <div xmlns="http://www.w3.org/1999/xhtml">
    <p><b>Example Call</b></p>
    <code>
GeoRegionQuery(dataSetId='AIRS', level='L2', version=None,
                startTime='2006/01/01T00:00:00', endTime='2006/02/01T00:00:00',
                latMin=-90., latMax=90., lonMin=-180., lonMax=180.,
                responseGroups='Large'
                )
    </code>
  </div>
</content>
<sf:callExample type="xml">
</entry>
```

Can embed HTML docs. in the 'content' tag.





Scast feed entry (3)

Can embed more metadata inside arbitrary tags, here 'sf:callExample'.

```
<sf:callExample type="xml">
  <process id="GeoRegionQuery">
    <inputs>
      <dataSetId>AIRS</dataSetId>
      <level>L2</level>
      <version>None</version>
      <startTime>2006/01/01T00:00:00</startTime>
      <endTime>2006/01/02T00:00:00</endTime>
      <latMin type="xs:float">-90.</latMin>
      <latMax type="xs:float">90.</latMax>
      <lonMin type="xs:float">-180.</lonMin>
      <lonMax type="xs:float">180.</lonMax>
      <responseGroups>Large</responseGroups>
    </inputs>
    <outputs>
      <resultSet type="sf:GeoRegionQueryResponse" />
    </outputs>
    <operator>
      <op>
        <binding>
          soap:http://sciflo.jpl.nasa.gov:8888/wsdl?http://sciflo.jpl.nasa.gov/2006v1/sf/EOSS
          ervices?geoRegionQuery
        </binding>
      </op>
    </operator>
  </process>
</sf:callExample>
</entry>
```





Feed Reader View

Firefox's Feed Reader:



Subscribe to this feed using Live Bookmarks

☐ Always use Live Bookmarks to subscribe to feeds

Subscribe Now

SciFlo Data Query/Access Services

On SciFlo node sciflo.jpl.nasa.gov

[GeoRegionQuery](#)

Space/time query and granule URL lookup services for multiple EOS satellite datasets at Levels L2/L3: AIRS, MODIS, MISR, GPS, and AERONET (ground network).

[FindDataById](#)

Translate a list of SciFlo data granule IDs to a list of on-line URLs.

For SOAP services, default links point to WSDL interface.





Browsing Service Casts

In Specialized Feed Reader / Aggregator

Signed in as bdwilson | [logout](#)

Feeds

Add Feed Remove

My Feeds

SciFlo Data Query/Access Services (df3)

- RGIS Base Imagery Service
- ADaM Services: Data mining and image process
- QuikSCAT Hurricane Ultra High-Res Winds
- GHRSSST AMSR-E SST
- ADaM Mining Web Services: Optimization Algori
- ADaM Mining Web Services: Image Processing
- ADaM Services: Data mining and image process
- HD - NASA's Jet Propulsion Laboratory
- NSIDC Atlas of the Cryosphere
- RGIS Base Imagery Service
- NSIDC Snow and Ice Cover
- ADaM Mining Web Services: Utility Services
- ADaM Mining Web Services: Texture Algorithms
- ADaM Mining Web Services: Pattern Recognition
- ADaM Mining Web Services: Domain Algorithms
- OOSTethys/MMI (Marine Metadata Interoperabilit
- Freeze/Thaw Earth System Data Record Data Vi

SciFlo Data Query/Access

[Open All](#) [Reading Pane](#) [Summary](#)

Title

Date ▾

GeoRegionQuery

Brian Wilson

3/13 1:32 am

Space/time query and granule URL lookup services for multiple EOS satellite datasets at Levels L2/L3: AIRS, MODIS, MISR, GPS, and AERONET (ground network).

FindDataById

Brian Wilson

3/13 1:32 am

Translate a list of SciFlo data granule IDs to a list of on-line URLs.

View in New Tab Go to Post

GeoRegionQuery

by Brian Wilson

Mar 13, 2008, 1:32 am

Example Call

```
GeoRegionQuery(dataSetId='AIRS', level='L2', version=None, startTime='2006/01/01T00:00:00', endTime='2006/02/01T00:00:00', latMin=-90., latMax=90., lonMin=-180., lonMax=180., responseGroups='Large' )
```

[Service interface description](#) [Server endpoint](#) [Service documentation](#) [SCast Source](#)





HTML View in Browser

SciFlo Data Query/Access Services

On SciFlo node **sciflo.jpl.nasa.gov**

Author: Brian Wilson, sciflo@sciflo.jpl.nasa.gov

Id: [uri:http://sciflo.jpl.nasa.gov/sciflo/v1/services/](http://sciflo.jpl.nasa.gov/sciflo/v1/services/)

GeoRegionQuery

Type: SOAP, Updated: 2008-03-13T01:32:02Z

Space/time query and granule URL lookup services for multiple EOS satellite datasets at Levels L2/L3: AIRS, MODIS, MISR, GPS, and AERONET (ground network).

Links: [Interface](#) [Service](#) [Documentation](#)

Example Call

```
GeoRegionQuery(dataSetId='AIRS', level='L2', version=None, startTime='2006/01/01T00:00:00',  
endTime='2006/02/01T00:00:00', latMin=-90., latMax=90., lonMin=-180., lonMax=180., responseGroups='Medium' )
```

FindDataById

Type: SOAP, Updated: 2008-03-13T01:32:02Z

Translate a list of SciFlo data granule IDs to a list of on-line URLs.

Links: [Interface](#) [Service](#) [Documentation](#)

Convert SCast to HTML view using XQuery or XSLT stylesheet





Readers Import List of SCasts

Feed Readers import OPML feed lists (OPML = outlining standard)

```
<opml version="1.0">
  <head>
    <title>Web Services Available from the ESIP Federation</title>
  </head>
  <body>
    <outline title="SciFlo" text="SciFlo">
      <outline text="Atmospheric Data Query and Access Services (primary)"
        title="Atmospheric Data Query and Access Services (primary)"
        type="rss"
        xmlUrl=
"http://sciflo.jpl.nasa.gov/sciflo/web/services/queryAndAccess.atom.xml"
        htmlUrl="http://sciflo.jpl.nasa.gov/" />
      <outline text="MISR Aerosol/Cloud Data Query and Access Services (primary)"
        title="MISR Aerosol/Cloud Data Query and Access Services (primary)"
        type="rss"
        xmlUrl=
"http://ldup05.larc.nasa.gov/sciflo/web/services/queryAndAccess.atom.xml"
        htmlUrl="http://sciflo.jpl.nasa.gov/" />
    </outline>
    <outline title="DataFed" text="DataFed">
    </outline>
    <outline title="LAITS/GMU" text="LAITS/GMU">
    </outline>
  </body>
</opml>
```



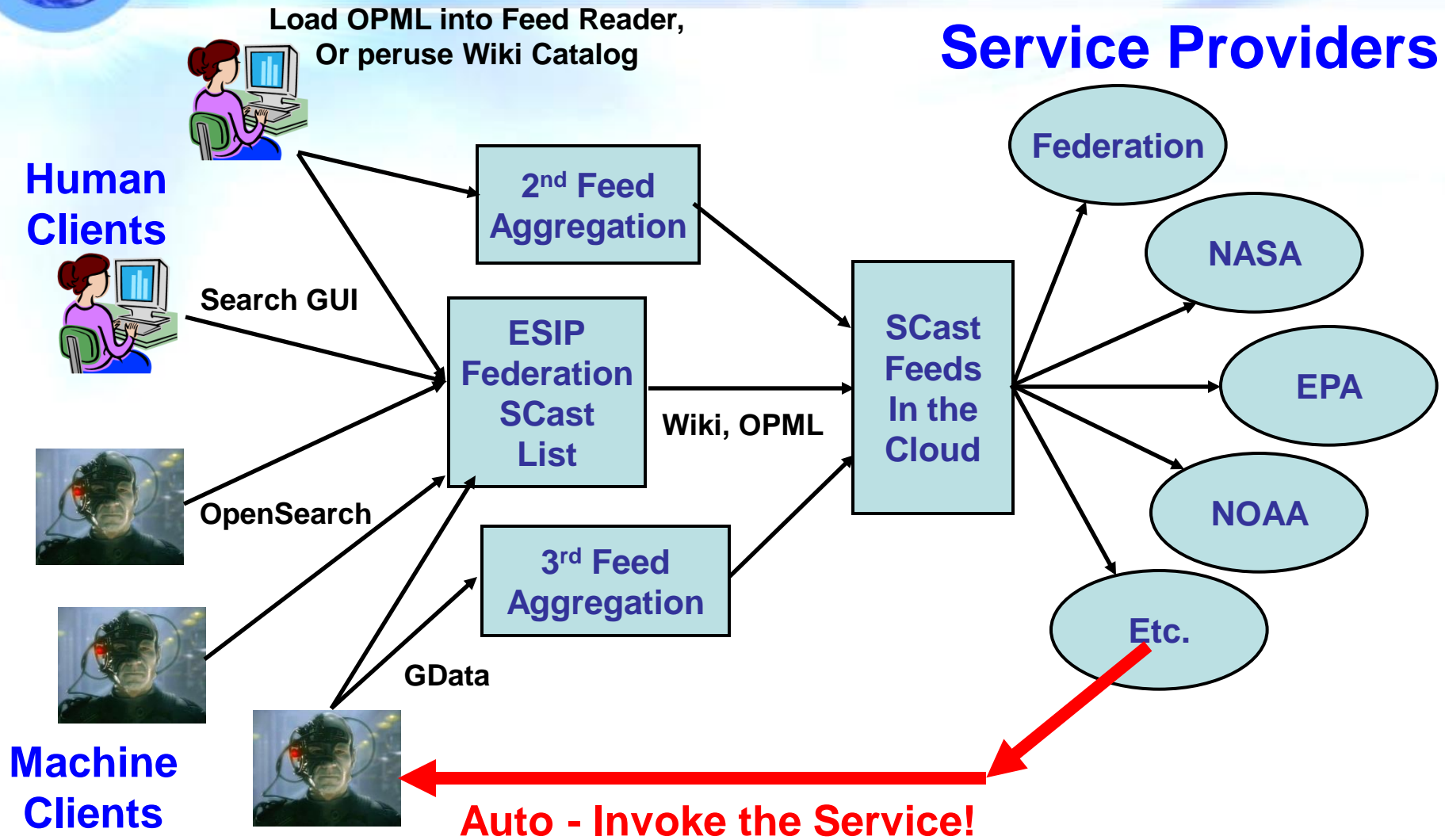


Create An SCast

- Simple service to create an “scast”
 - <http://tinyurl.com/createSCast>
 - Select REST or SOAP version of service
 - Hit execute
 - Result is scast (click download link at bottom of results page)
 - If your browser auto-displays Atom feeds, you’ll see it.
 - Use “page source” to download XML text
- Aggregating “scasts”
 - Send me the URL pointing to your scast feed
(Brian.Wilson@jpl.nasa.gov)
 - I will aggregate them onto a ESIP Federation page, and an OPML file that can be imported into feed readers
 - If you have suggestions or questions about some fields, email me.



SCast Distributed Architecture





Interface, Interface, Interface

- You can never be too simple, or have too many interfaces.
- SCast feed links on ESIP Federation wiki page
 - Any service provider can add their ad feed
 - Scasts are auto-aggregated from wiki into OPML file
 - OPML catalog can be XSL-transformed into HTML viewable catalog
- Opensearch / GoogleData interface
 - Everything is search, all results are feeds.
 - Search/filter scasts using REST call
- Human GUI for feed search
 - Many feed readers
- What else?





Open Search Interface

- Everything is search, all results are feeds!
- Discover Service Bundles (scast entries)
 - /scast/-/services?q=parameter+subsetting
&startIndex=1&count=200&format=atom
 - Returns scast entries satisfying query keywords
- Discover Services by Provider
 - /scast/-/services/providerId?q=space+reprojection
&startIndex=1&count=200&format=atom
 - Returns scast entries only from that provider
- OpenSearch (GData) Features
 - GoogleData standard uses and extends OpenSearch
 - Search aggregators auto-handle Atom feed, traverse result sets
 - Google already crawls Atom feeds, making them searchable





Future Possibilities

- Describe owner of scast
 - <scast:provider> tag, or link
 - <link rel="scast:provider" />
 - Could also reuse Dublin Core tags, <dc:publisher>
- Categories (contents of <category> tag)
 - Just text keywords?
 - Develop services taxonomy?
- Specify availability and maturity of the feed
 - Experimental, or operational 24x7
- Add tag to point to other scasts
 - Create a “web ring” of scasts
 - Could point to related or recommended alternative scasts
 - <link rel="scast:nextScast" />
 - <link rel="scast:relatedServices" />
- Who is allowed to use the service? Permissions?
 - <link rel="scast:registerHereForUse" />





Issues / Future

■ Solicit Feedback

- Missing service types?
- Categorizing services – taxonomy?

■ Adoption

- Twist arms RIGHT NOW to author scasts
- Evangelize within Federation
- Author feeds for people, and send them the draft

■ Make free tools available

- HTML “views” of scast and OMPL using stylesheets
- Custom feed reader / filter

■ Promulgate as wider standard

- Scasts for everyone, simpler than UDDI
- Generate scast of all services registered in ECHO (??, info. for all three links might not be present)





Summary

- Reduce advertising services to publishing an “scast” (Atom) feed
 - Feed readers already aggregate & filter feeds
 - Search also available
- Decentralized repository with auto-aggregation
 - Service providers control publishing of advertisements
 - Scast discoverable in the cloud
 - Publish scast to Federation list by creating wiki link
 - Script auto-aggregates to list of scasts (OPML file)
- Scast contains machine-readable metadata to categorize services and enable auto-invocation
 - Links to interface, service endpoint, human documentation
- So simple it will be adopted (unlike previous attempts).
 - Make free tools available
 - Evangelize





Required Lab

- **Let's author some scasts!!!**
- **Download example files:**
 - http://sciflo.jpl.nasa.gov/scast/SciFlo_GranuleQuery_SOAP.scast.atom
 - http://sciflo.jpl.nasa.gov/scast/AIRS_NRT_WMS.scast.atom
- **Author a ServiceCast**
 - Start with example scast
 - What is the service type? Need a new type?
 - Lookup URL's to your interface (WSDL, WADL, GetCapabilities), service endpoint, and docs. for humans
 - For feedback or authoring help: Brian.Wilson@jpl.nasa.gov
- **Explore capabilities of your favorite Feed Reader**
 - How does it format the scast?
 - Can you filter on text keywords?
- **Publish your ServiceCast on the Federation wiki**

